

EINGABENPROGRAMM Nr. 1

11.0 F

Funktionen:

Liest Dezimalzahlen, wandelt sie ins Dualsystem um und speichert sie in einer vorgegebenen Zelle. Die Kommalage von Dezimal- und Dualzahl sowie der Speicherplatz werden für jede Zahl durch ein Schlüsselwort angegeben.

Eingabe:

Für jede Zahl ein Codewort der Form

P ± qq mmmmm

und die einzulesende Zahl. Ein Nullwort ruß der letzten Dezimalzahl folgen.

Ausgabe:

Die Dezimalzahlen werden umgewandelt und in den vorgegebenen Zellen gespeichert.

Befehlsfolge:

a	R (L ₀ + 8)	} 11,0F
a+1	U L ₀	
a+2	u.s.w.	

Ausgang:

Nach Einlesen des Schlußzeichens erfolgt ein Sprung zur Zelle a+2.

Codewort:

Das Codewort hat die Form

P ± qq mmmmm

Dabei bezeichnet P die Lage des Kommas der Dezimalzahl, $\pm qq$ die Anzahl der Zwischenstellen der Dualzahl und $mmmm$ die Adresse der Zelle, die die Zahl aufnehmen soll. die Zahl P zählt die Ziffern, die hinter dem Komma stehen; es muß sein: $0 \leq P \leq 9$. Die Zahl q ist durch P eingeschränkt. Die untenstehende Tabelle gibt obere und untere Schranken für vorgegebenes P . Alle Zahlen P , q und m werden dezimal eingegeben.

Zur Darstellung einer Zahl x mit einem gegebenen q ist

$$x \leq 2^q$$

notwendig. Ist q jedoch zu groß, so wird die umgewandelte Zahl Fehler enthalten, die u. U. nicht genügend Dualstellen zur Verfügung stehen.

Tabelle der P und q

P	Max q	Min q	K	m
0	+47	+02	+30	+24
1	+43	-02	+26	+20
2	+40	-05	+22	+17
3	+37	-08	+18	+14
4	+34	-11	+16	+10
5	+31	-14	+13	+07
6	+28	-17	+10	+04
7	+24	-21	+06	+00
8	+21	-24	+03	-03
9	+17	-28	+00	-06

Die beiden letzten Spalten geben das maximale (K) und das minimale (m) q an, für das exakte Umwandlung möglich ist.

Zeitbedarf:

In einer Minute werden 20 bis 25 Zahlen umgewandelt.

Fehler:

Die letzte Dualziffer der umgewandelten Zahl kann fehlerhaft sein.

Speicherbedarf:

Der Plan benötigt 192 Plätze für Befehle und Konstanten. Die Zellen 03, 04, 45, 52 und 54 - 57 von Spur 63 werden als Zwischenspeicher benutzt.

Wart:

Ist der Betrag von X größer als 2^q , so stoppt die Maschine bei $L_0 + 0234$.

Beispiel:

Die Zahlen +96.40236, -0.00000597 und +330000 sind mit $q = 7$, -14 bzw. +30 in den Zellen 6234, 2363 bzw. 2100 zu speichern.

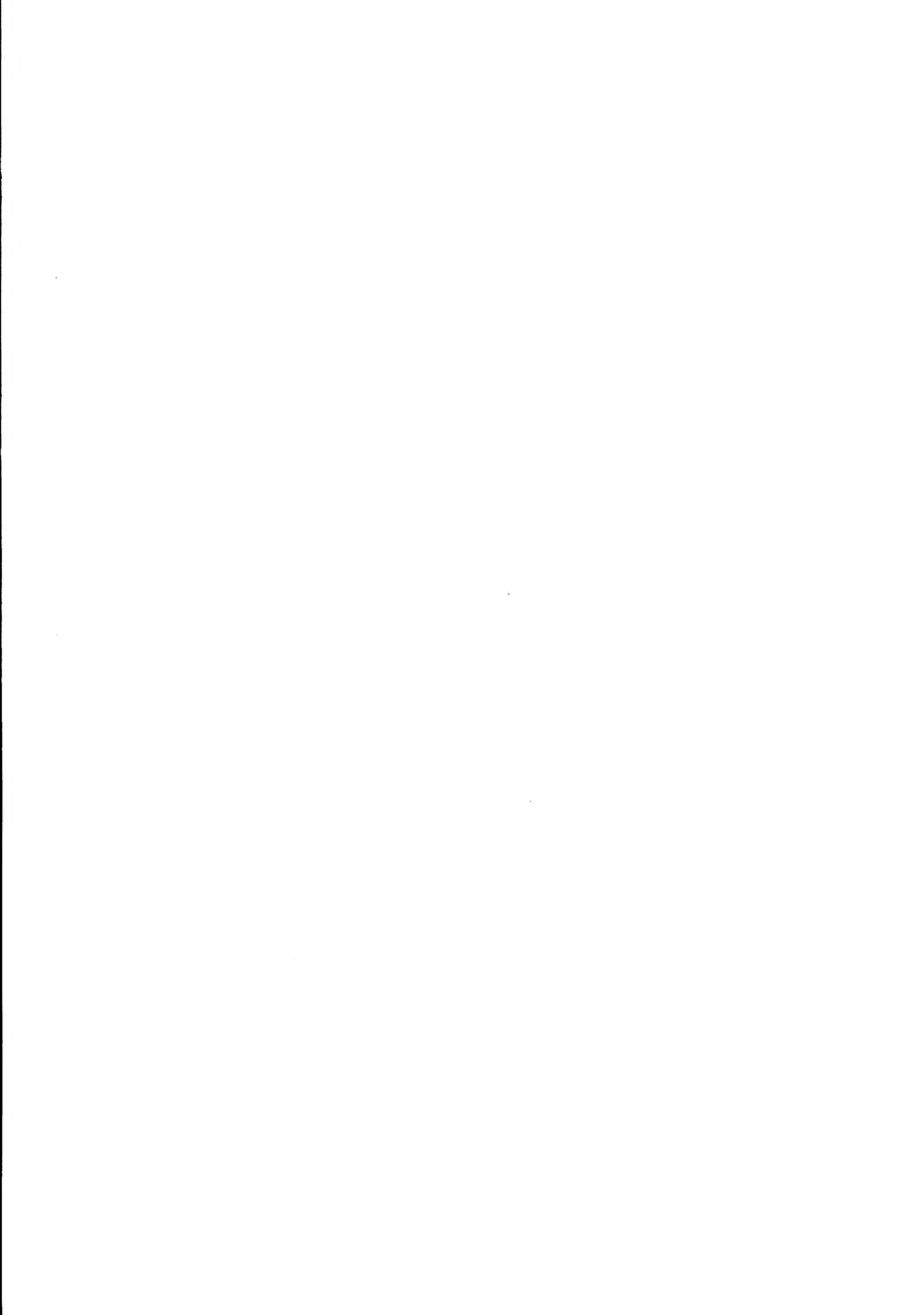
5+07	6234'	+9640236'
9-14	2363'	-0000597'
1+30	2100'	+3300000'

Bemerkungen:

Alle Ziffern des Codewortes müssen gelocht werden, so daß dieses immer aus acht Stellen besteht. Führungsnullen bei positiven Zahlen können fortfallen, bei negativen müssen jedoch Vorzeichen und sieben Stellen gelocht werden. Für die letzte Zahl des Beispiels könnte man auch schreiben

0+302100' 330000'

Man prüfe sorgfältig, ob der letzten Zahl ein zusätzlicher Stoppcode folgt.



J2-11.0R

u0115' e0044' xp0050' xi0000' xh6554' t0022'
u000' s0029' t0000' a0145' n0132' e0255'
xh6555' u0017' m0136' u0044' xz0030' e0060'
m0140' xa6355' xh6356' u0025' r0044' u0010'
xz0032' e0240' m0262' xa6556' u0031' ,0000002' 2' y0260'
s0247' t0220' e0213' xc6557' xs6557' m0059'
xh6503' xb6554' e0219' u0014' b0121' u0100'
a0000' u0040' xh6504' m0133' a0113' y0255'
y0222' b0016' xs6503' xs6504' xp0012' t0227'
r0114' u0061' u0254' ,0000002' 20000' 3j5j5j0' a0226'
c0254' xi0000' n0029' e0257' xh6552' s0239'
t0259' s0141' t0042' b0245' y0144' xb6552'
e0240' xh6554' u0116' z0240' u0000' a0050'
e0258' m0146' xa6354' xh6555' u0124' z0114'
' m0148' e0139' m0147' xa6555' xh6556'
e0143' m0215' u0154' ,0000003' 4' 20000000'
xa6556' u0144' xz0032' b0123' u0241' ,0000041'
lq01wq00' k0000000' lwwwwwq' lwwq0000'
u0000' 2' k0000000' g2000000' (wwwwwwq' 40000000'
20000000' 10000000' 80000000' 40000000' 20000000'
10000000' 8000000' 4000000' 2000000' 1000000'
80000' 40000' 20000' 10000' 8000' 4000' 2000' 1000'
800' 400' 200' 100' 80' 40' 20' 10' 8' www0000' 95680000' xc6552'
xs6552' u0114' ,0000001' (0000000' a0263'
u0037' a0000' m0000' u0260' ' a0148' xc6556'
r0114' u0137' u0222' ,0000005' ' ' 5wkkwj'
a0000' a0000' u0260' ,0000004' 5wwwwwwq' lqlqlq0'
2' wwww00' xs6556' u0244' z0216' c0223' u0063'
,0000013' lwwwwwq' 1000000' (wwwwwwq' 50000000'
32000000' lw400000' 13680000' j350000'
7f12000' 4j4g400' 2wfw060' lkjx650' ' xc6545'
e0000' u0000' ,0000002' f0000000' 1000000'
.0000000'